C □□□□□□□□

□□

□□

□□□
□□□□□□□

□□□□□□ C □□□□□□□□□□□□□□□□□□ C □□□□□□□ C □□□□□□□
Windows□Linux□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□ C □□□□□□□□□□□ C □□□□□□□
□□□ C

□□□
□□□□□ QQ □□□□□□□□

C □□□□□□□□□□□□□□ C □□□□□□□□□□□□□□□□□□□□□□□□□□
□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□
□□□□□□□□□□□□□□□□□ Windows XP SP3□□□□□□□□□□□□□□□
□□□□□□□□□□□□□ Visual C++ 6.0

[illegible]

[illegible]

[illegible] [illegible]

[illegible] VC 6.0 [illegible] MFC [illegible] VC [illegible] MFC [illegible]

1.1

[illegible] C [illegible]

[illegible] Hello World [illegible]

```
#include <stdio.h>
void main()
{
```

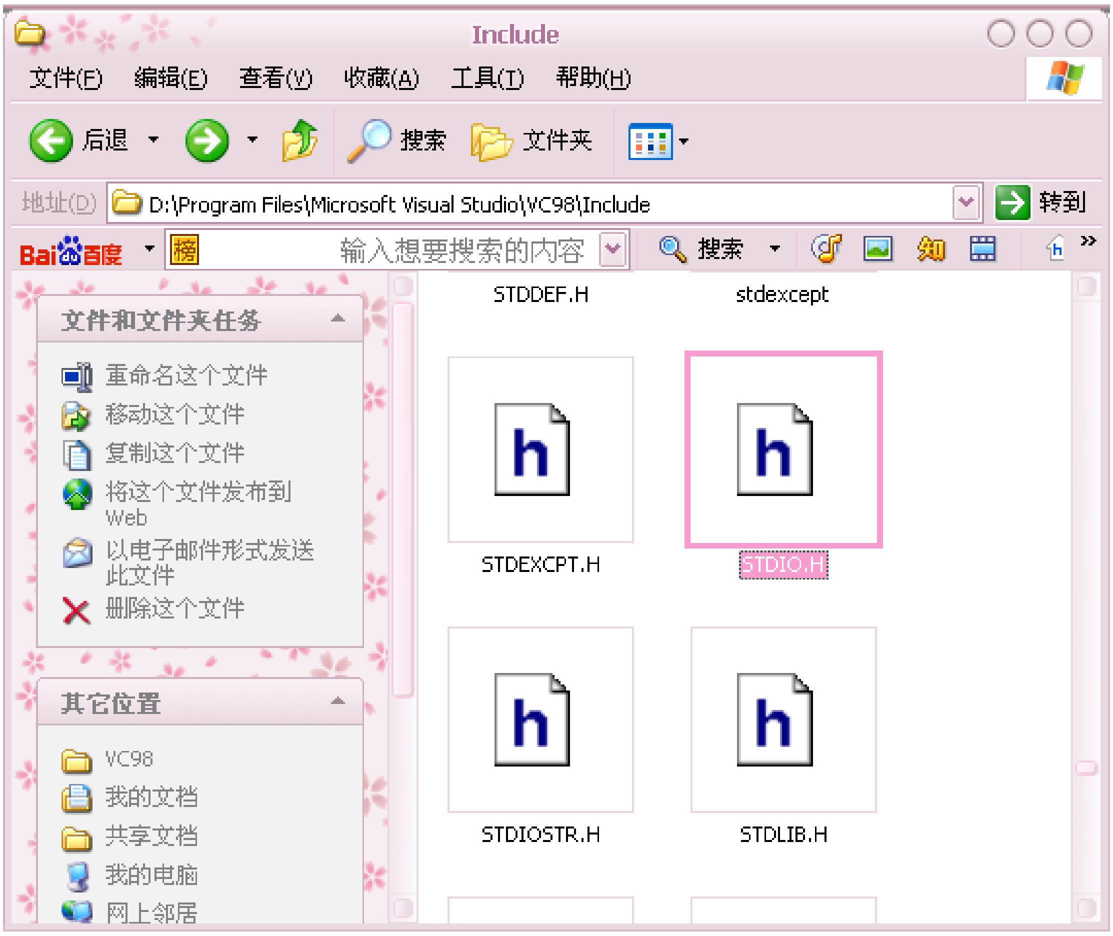
```
    printf("Hello World!");
 }
```

□□□□□□□□□□C□□□□□□□□C□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

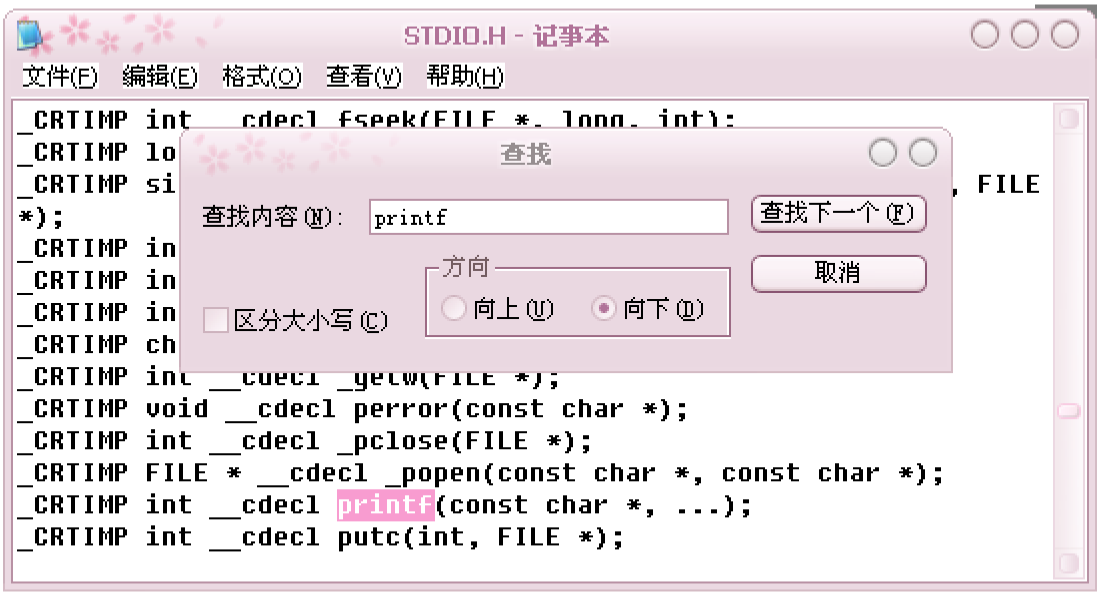
#include

<stdio.h>□□□□□stdio.h□□□□□□□□□□□□□□□stdio.h□□□□□□□□□□□□□

□□□□VC□□□□□□□□□□



□□□□□□printf□□□□□□stdio.h□□□□□□□□□□□□□□□□□□□□□□□□□□



[illegible]

[illegible] stdio.h [illegible] printf [illegible] stdio.h [illegible] printf [illegible]

[illegible]“Hello World!”,[illegible] Hello World!

```
#include <windows.h>
void main()
{
    MessageBox(NULL," Hello World!","[illegible]",MB_OK);
}
```

[illegible]

[illegible] Hello World[illegible]“[illegible]”[illegible]

[illegible]

[illegible]

[illegible]

[illegible] MessageBox [illegible]

[illegible] C [illegible] VC [illegible]

[illegible] Windows API [illegible]

Windows API [illegible] MSDN [illegible]

[illegible] Library[illegible]

[illegible] MSDN:http://msdn.microsoft.com/library/zh-cn/

[illegible] MSDN:http://msen.micorsoft.com/library/en-us/

[illegible]

[illegible]

[illegible]

```
#include “stdio.h”
int i=0,j=0;
for(i=1;i<10;i++)
    for(j=1;j<i+1;j++)
        printf(“%d*%d=%d \t”,j,i,j*i);
    printf(“\n”);
```

[illegible] javascript [illegible]

[illegible] MSDN [illegible]

1.2 [illegible]

[illegible] MessageBox [illegible]

[illegible]

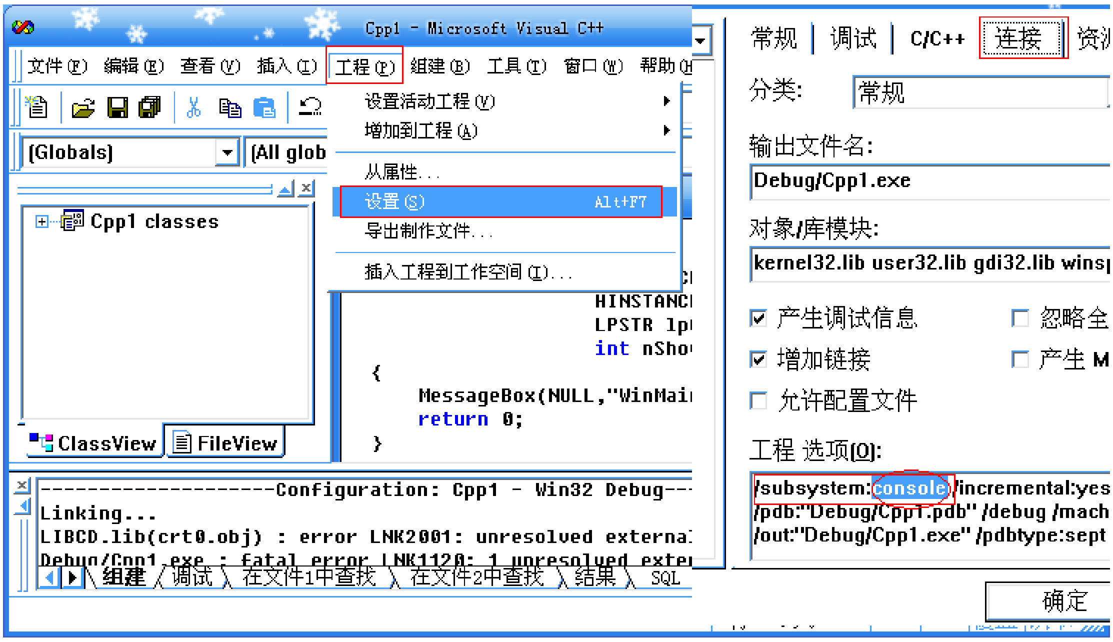
main [illegible] VC 6.0 [illegible] VC 6.0 [illegible] WinMain()

[illegible]

```
int WINAPI WinMain(
     HINSTANCE hInstance,
     HINSTANCE hPrevInstance,
     LPSTRlpCmdLine,
     int nCmdShow
    );
```

[illegible] main [illegible] main [illegible] main [illegible] WinMain [illegible] VC 6.0 [illegible]

□□□



□□ □“□□”→“□□”→“□□”□□“□□□□”□□ console □□ windows □□□□□□□□□□□□□□□□□□□□□□□□□□□ C □□□□□□□□□□□□□□□□Console □□□□□□□□□□□□□□□ mian □□ Windows□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
#include "windows.h"


int WINAPI WinMain(HINSTANCE hInstance,
                    HINSTANCE hPreInstance,
                    LPSTR lpCmdLine,
                    int nShowCmd)
{
    MessageBox(NULL,"WinMain □□□□□□□","WinMain",MB_OK);
```

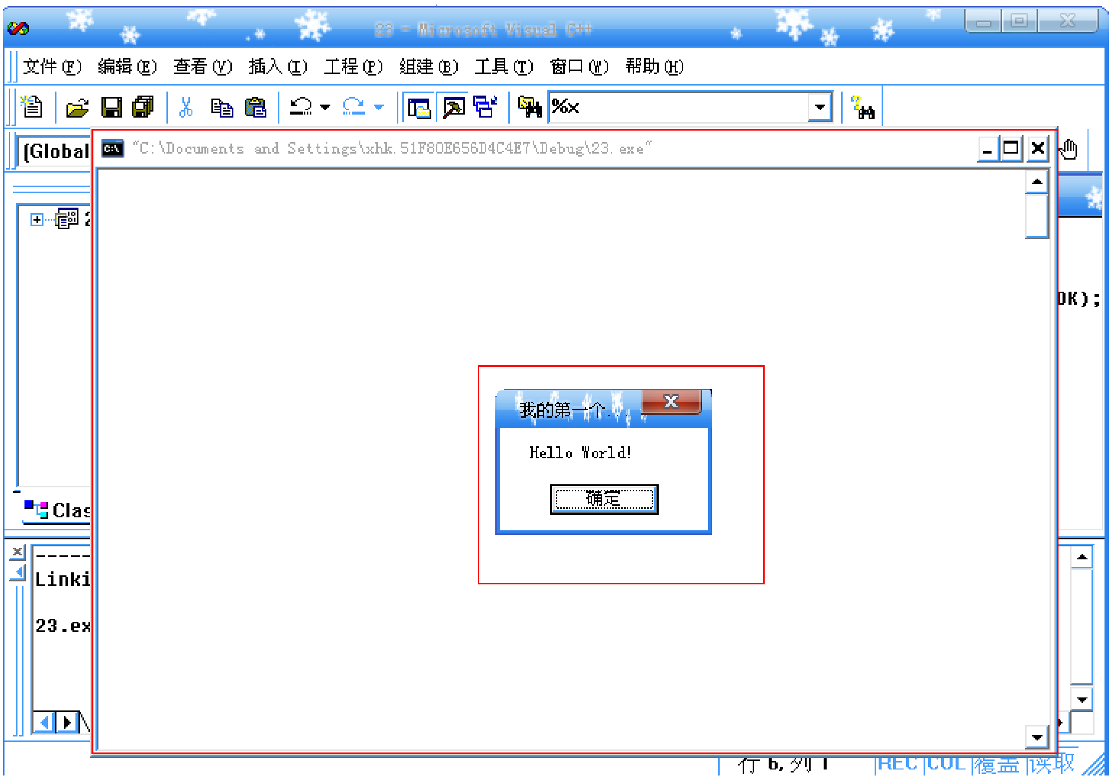
```
    return 0;
}
```

□□□□□:

MessageBox(NULL," Hello World!","□□□□□□□□□□",MB_OK);

□□□□□□□□□□□□□□□□□

```
#include “windows.h”
void main()
{
    MessageBox(NULL," Hello World!","□□□□□□□□□□",MB_OK);
}
```



□□□□□□□□□□□cmd □□□□□□main □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

winmain □□

[illegible] WinMain() [illegible] 4 [illegible]

[illegible]

hInstance [illegible]

hPreInstance [illegible]

[illegible] 32 [illegible] NULL [illegible]

lpCmdLine [illegible]

GetCommandLine [illegible]

nCmdShow [illegible] MSDN

[illegible] (HANDLE) [illegible]

[illegible] Windows [illegible]

Windows [illegible] ID [illegible] ID [illegible] ID [illegible]

[illegible]

[illegible]

[illegible]

[illegible] Windows [illegible]

[illegible] C [illegible]

[illegible] C [illegible]

[illegible] C [illegible]

[illegible]

```
struct People
{
    int age;//[illegible]
    char sex[2];//[illegible]
    int height;//[illegible]
……
}
```

[illegible]People zhangsan;[illegible]zhangsan[illegible]

[illegible]172[illegible]zhangsan.height[illegible]

[illegible]JAVA[illegible]java[illegible]

[illegible]

```
public class People
{
    public int age;
    public string sex;
    public height;
    ……
}
```

[illegible]

[illegible]

[illegible] WinMain [illegible]

HINSTANCE [illegible] HANDLE [illegible]

HANDLE [illegible] typedef PVOID HANDLE; [illegible] PVOID [illegible]

typedef void *PVOID; [illegible] PVOID [illegible] void [illegible]

[illegible]

[illegible] msdn [illegible]

[illegible] LPSTR [illegible]

[illegible] msdn [illegible]

[illegible] WinMain [illegible]

[illegible]

1.3 [illegible]

[illegible]

[illegible]

[illegible] MessageBox() [illegible] MessageBox() [illegible]

```
Int MessageBox(HWND hWnd,
               LPCTSTR lpText,
```

LPCTSTR lpCaption,

UINT uType);

□□□□

hWnd □□□□□□□□□□□□□□□□NULL□□□□□□□□□□□□□□□□

lpText □□□□□□□□□□□

lpCaption □□□□□□□□□

uType □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□

http://msdn.microsoft.com/en-us/library/ms645505(VS.85).aspx

□□□C□□□□□□□□□□C□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Windows□□□□□□□□□□□□□□□□□□□

1. □□□□□□□□□□□□□□Windows□□□□□□□□□wndclassex□□□□□□□□□□□□□□□□□□□WNDCLASSEX(□□□)□□□□□□□□□□□□□□(http://msdn.microsoft.com/en-us/library/ms633577(VS.85).aspx)□

```
typedef struct {
    UINT cbSize;
    UINT style;
```

```
    WNDPROC lpfnWndProc;
    int cbClsExtra;
    int cbWndExtra;
    HINSTANCE hInstance;
    HICON hIcon;
    HCURSOR hCursor;
    HBRUSH hbrBackground;
    LPCTSTR lpszMenuName;
    LPCTSTR lpszClassName;
    HICON hIconSm;
} WNDCLASSEX, *PWNDCLASSEX;
```

□□□□

cbSize □□ sizeof(WNDCLASSEX),□□□ GetClassInfoEx □□□□□

□□□□□

style □□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□

lpfnWndProc [illegible]

[illegible]

cbClsExtra [illegible] 0[illegible]

cbWndExtra [illegible] 0.[illegible] WNDCLASSEX [illegible]

[illegible] DLGWINDOWEXTRA

hIcon [illegible] NULL[illegible]

hCursor [illegible] NULL[illegible]

hbrBackground [illegible]

[illegible] HBRUSH [illegible]

- COLOR_ACTIVEBORDER
- COLOR_ACTIVECAPTION
- COLOR_APPWORKSPACE
- COLOR_BACKGROUND
- COLOR_BTNFACE
- COLOR_BTNSHADOW
- COLOR_BTNTEXT
- COLOR_CAPTIONTEXT
- COLOR_GRAYTEXT
- COLOR_HIGHLIGHT

- COLOR_HIGHLIGHTTEXT
- COLOR_INACTIVEBORDER
- COLOR_INACTIVECAPTION
- COLOR_MENU
- COLOR_MENUTEXT
- COLOR_SCROLLBAR
- COLOR_WINDOW
- COLOR_WINDOWFRAME
- COLOR_WINDOWTEXT

lpszMenuName [illegible] NULL [illegible] id [illegible] MAKEINTRESOURCE [illegible] NULL,[illegible]

lpszClassName [illegible]

hIconSm [illegible]

[illegible] WNDCLASSEX [illegible]

RegisterWindowEx(&wndclassex)[illegible]

[illegible]

2. □□□□

□□□□□□□□□□□ CreateWindowEx()□□□□□□□□□

CreateWindowEx □□□□□□□:

```
HWND CreateWindowEx(

    DWORD dwExStyle,

    LPCTSTR lpClassName,

    LPCTSTR lpWindowName,

    DWORD dwStyle,

    int x,

    int y,

    int nWidth,

    int nHeight,

    HWND hWndParent,

    HMENU hMenu,

    HINSTANCE hInstance,

    LPVOID lpParam
```

);

[illegible]

dwExStyle[illegible]

WS_EX_ACCEPTFILES[illegible]

WS_EX_APPWINDOW[illegible]

WS_EX_CLIENTEDGE[illegible]

WS_EX_CONTEXTHELP[illegible]

WS_EX_CONTROLPARENT[illegible] Tab [illegible]

WS_EX_DLGMODALFRAME[illegible] dwStyle [illegible] WS_CAPTION [illegible]

WS_EX_LEFT[illegible]

WS_EX_LEFTSCROLLBAR[illegible] Hebrew[illegible]Arabic[illegible] reading order alignment [illegible]

WS_EX_LTRREADING[illegible] LEFT [illegible] RIGHT[illegible]

WS_EX_MDICHILD[illegible] MD [illegible]

WS_EX_NOPATARENTNOTIFY[illegible] WM_PARENTNOTFY [illegible]

WS_EX_OVERLAPPED□

WS_EX_CLIENTEDGEWS_EX_WINDOWEDGE □□□□

WS_EX_PALETTEWINDOW□WS_EX_WINDOWEDGE,

WS_EX_TOOLWINDOW □ WS_WX_TOPMOST □□□□□ WS_EX_RIGHT:□

□□□□□□□□□□□□□□□□□□□□□

WS_EX_RIGHTSCROLLBAR□□□□□□□□□□□□□□□□□□□□□

□□

WS_EX_RTLREADING□□□□□□□□□ Hebrew□Arabic□□□

□□□□□□□□□reading order alignment□□□□□□□□□□□□□□□□□□RIGHT □

LEFT □□□□□□□□

WS_EX_STATICEDGE□□□□□□□□□□□□□□□□ 3 □□□□□□

WS_EX_TOOLWIDOW□□□□□□□□□□□□□□□□□□□□□□

WS_EX_TOPMOST□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ L□□□

□□□□□□□□□□□ SetWindowPos □□□□□□□□□□□

WS_EX_TRANSPARENT□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□

IpClassName: □□□□□□□

lpWindowName:[illegible]

dwStyle:[illegible]

x:[illegible]

y:[illegible]

nWidth:[illegible]

nHeight:[illegible]

hMenu[illegible]

hInstance[illegible]

lpParam:[illegible] WM_CREATE [illegible]

[illegible] NULL[illegible]

[illegible] GetLastError [illegible]

3. [illegible]

[illegible]

[illegible] ShowWindow[illegible]

```
BOOL ShowWindow(

    HWND hWnd,//[illegible]

    int nCmdShow //[illegible]

        );
```

[illegible] CreateWindowEx [illegible]

ShowWindow()[illegible]

[illegible] UpdateWindow()[illegible]

[illegible]

BOOL UpdateWindow(HWND hWnd);

[illegible]

[illegible] UpdateWindow [illegible] WM_PAINT [illegible]

[illegible] WM_PAINT [illegible] API

[illegible] WM_PAINT [illegible]

[illegible]

hWnd [illegible]

[illegible]

[illegible]

[illegible]

[illegible] Windows [illegible]

[illegible] Windows [illegible]

[illegible]

[illegible] ok [illegible] T [illegible]

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□cangsanbujin@126.com□

□□□□□□□□□□□□□□□□□□□□□□

```
#include <windows.h>


//□□□□

LRESULT WINAPI WinProc(HWND hWnd,UINT Msg,WPARAM

wParam,LPARAM lParam)

{

    switch(Msg)//□□□□□□□□□□□□□□□□□

    {

    case WM_DESTROY://□□□□□□□□□□□□

        PostQuitMessage(0);//□□□□□□□□□□□□□□□□□□□□

        return 0;//□□□□

    }

    return DefWindowProc(hWnd,Msg,wParam,lParam);

}
```

```
//□□□

int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE

hPrevInstance,LPSTR lpCmdLine,int nShowCmd)

{

    char *cName = "myWindow";

    WNDCLASSEX wc;

    HWND hWnd;

    MSG Msg;


    wc.cbClsExtra = 0;

    wc.cbWndExtra = 0;

    wc.cbSize = sizeof(WNDCLASSEX);

    wc.hbrBackground =

(HBRUSH)GetStockObject(WHITE_BRUSH);//□□□□□□□□□□□□□□□□□□

□□□□NULL□□□□□□□□

    wc.hCursor = NULL;//□□□

    wc.hIcon = NULL;//□□□

    wc.hIconSm = NULL;//□□□
```

```
    wc.hInstance = hInstance;//□□□□□□□□hInstance □□□□□□□□
    wc.lpfnWndProc = WinProc;//□□□□□□□□□□□□
    wc.lpszClassName =(LPSTR)cName;//□□□□□□□
    wc.lpszMenuName = NULL;//□□□□□□□
    wc.style = CS_HREDRAW | CS_VREDRAW;

    RegisterClassEx(&wc);//□□□□□□

    hWnd = CreateWindowEx(WS_EX_CLIENTEDGE,cName,"□□□□□
□□",WS_OVERLAPPEDWINDOW,
        200,100,600,400,NULL,NULL,hInstance,NULL);//□□□□□□□□□
□"□□□□□□□"
    if(hWnd == NULL)
    {//□□□□
        MessageBox(NULL,"There's an Error","Error
Title",MB_ICONEXCLAMATION|MB_OK);
        return 0;
    }
```

```
    ShowWindow(hWnd,nShowCmd);//□□□□
    UpdateWindow(hWnd);

    //□□□□□□□□□□□□□□□□□□□□□□□□
        while(GetMessage(&Msg,NULL,0,0))
    {
        TranslateMessage(&Msg);//□□□□
        DispatchMessage(&Msg);//□□□□
    }
    return Msg.message;
}
```

□□□□□□□□□□□□□□□□□□□□□□□□□□

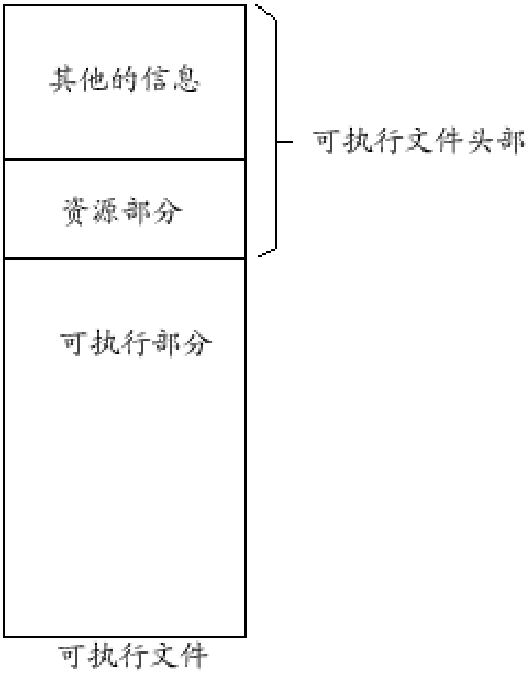
[illegible] VC [illegible] WNDCLASSEX [illegible]

1.4 [illegible]

[illegible] [illegible]

[illegible]

[illegible] Windows [illegible] cpu [illegible]



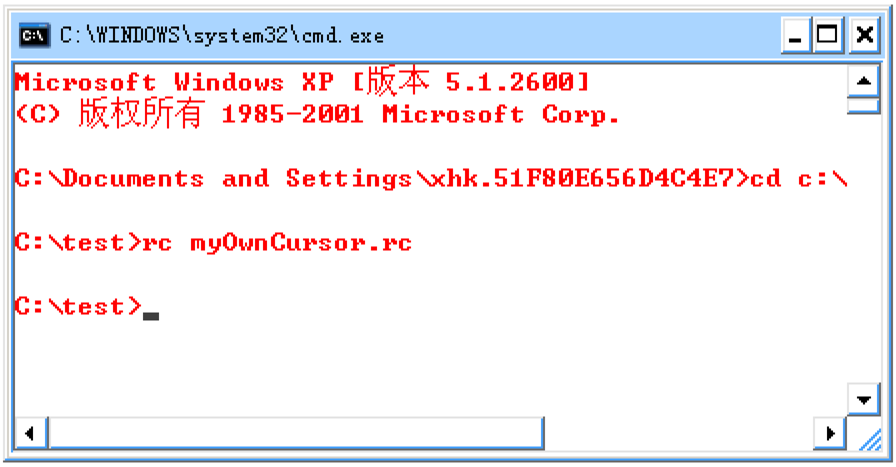
[illegible]

[illegible] rc [illegible] C [illegible] Rc.exe [illegible] res [illegible] res [illegible] obj [illegible] exe [illegible] cur [illegible] ani [illegible] myOwnCursor.rc

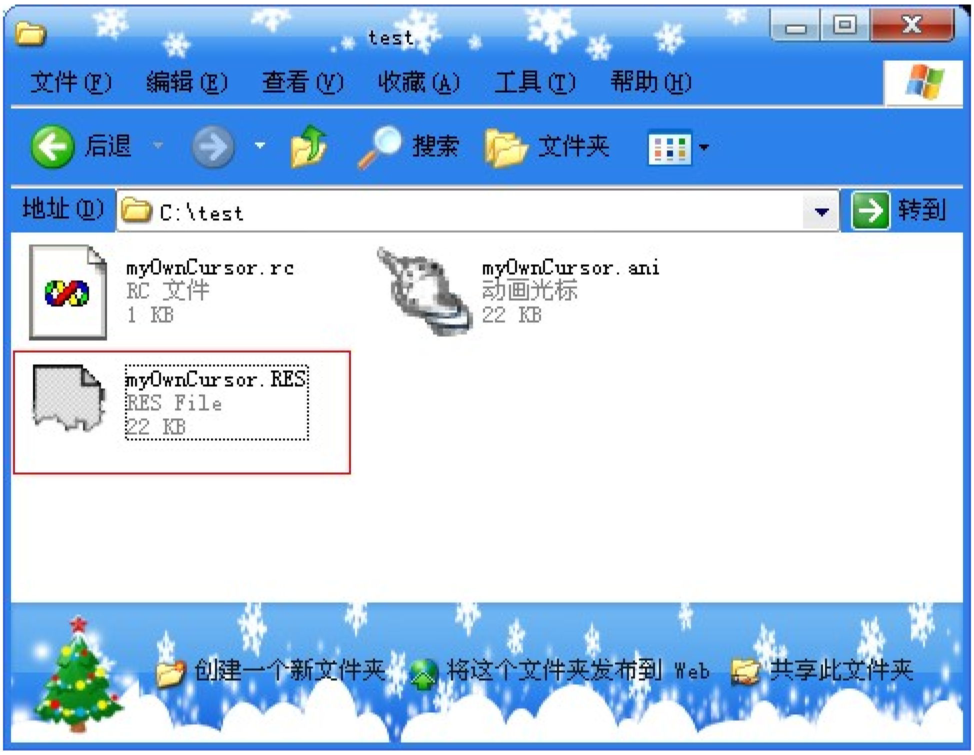
```
//myOwnCursor.rc written by xhk 2009.3.1
#include   <resource.h>   //[illegible]
#define    CUR    0x1000                    //[illegible] ID [illegible] id
```

CUR CURSOR "myOwnCursor.ani" //□□□□□□

□□□□□□□□□□□□□□□□rc.exe □□□□rc myOwnCursor.rc □□□□



□□□□□□□□□□□myOwnCursor.RES □□□□□□□□□□□□□□□



□□□□□□□□□□□□□□□□□□□□myOwnCursor.c □□□□□□□□□□□□□

□□□□□□□□□□□□□

```
#include <windows.h>

#define CUR 0x1000 //□□□□□□ id

//□□□□
LRESULT WINAPI WinProc(HWND hWnd,UINT Msg,WPARAM wParam,LPARAM lParam)
{
    switch(Msg)//□□□□□□□□□□□□□□□□□
    {
    case WM_DESTROY://□□□□□□□□□□□□
        PostQuitMessage(0);//□□□□□□□□□□□□□□□□□□□□
        return 0;//□□□□
    }
    return DefWindowProc(hWnd,Msg,wParam,lParam);
}

//□□□
```

```
int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine,int nShowCmd)
{
    char *cName = "myWindow";
    WNDCLASSEX wc;
    HWND hWnd;
    MSG Msg;

    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;
    wc.cbSize = sizeof(WNDCLASSEX);
    wc.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);//□□□□□□□□□□□□□□□□□□□□□□NULL□□□□□□□□
    wc.hCursor = LoadCursor(hInstance,MAKEINTRESOURCE(CUR));//□□□□□□□□□□□□
    wc.hIcon = NULL;//□□□
```

```
    wc.hIconSm = NULL;//□□□

    wc.hInstance = hInstance;//□□□□□□□hInstance □□□□□□□

    wc.lpfnWndProc = WinProc;//□□□□□□□□□□□

    wc.lpszClassName =(LPSTR)cName;//□□□□□□

    wc.lpszMenuName = NULL;//□□□□□□

    wc.style = CS_HREDRAW | CS_VREDRAW;


    RegisterClassEx(&wc);//□□□□□


    hWnd = CreateWindowEx(WS_EX_CLIENTEDGE,cName,"□□□

□□□□",WS_OVERLAPPEDWINDOW,

        200,100,600,400,NULL,NULL,hInstance,NULL);//□□□□□□□

□□"□□□□□□"

    if(hWnd == NULL)

    {//□□□□

        MessageBox(NULL,"There's an Error","Error

Title",MB_ICONEXCLAMATION|MB_OK);

        return 0;
```

```
    }
    ShowWindow(hWnd,nShowCmd);//□□□□
    UpdateWindow(hWnd);

    //□□□□□□□□□□□□□□□□□□□□□□□□□
     while(GetMessage(&Msg,NULL,0,0))
    {
        TranslateMessage(&Msg);//□□□□
        DispatchMessage(&Msg);//□□□□
    }
    return Msg.message;
}
```

□ vc □□□□ myOwnCursor.obj,□ myOwnCursor.obj □

myOwnCursor.res □□□□□□□□□□□□□□□□□□□□□□ □□□□□□linke

kernel32.lib user32.lib gdi32.lib /subsystem:windows

myOwnCursor.obj myOwnCursor.res □□□□□□□□

myOwnCursor.exe.□□□□□□□;

[illegible]

[illegible] C [illegible]
Windows [illegible]

[illegible]

1.5 [illegible]“Hello World”

[illegible]

[illegible]windows [illegible]

Windows [illegible]

windows [illegible]

[illegible]VB [illegible]VB [illegible]

[illegible]JAVA [illegible]JAVA [illegible]

[illegible]JAVA[illegible]VB[illegible]VB[illegible]

[illegible]VB[illegible]

[illegible]C[illegible]C[illegible]C[illegible]C++

[illegible]

[illegible]

[illegible]

[illegible]

```
while(GetMessage(&Msg,NULL,0,0))
{
    TranslateMessage(&Msg);//[illegible]
    DispatchMessage(&Msg);//[illegible]
}
```

Windows [illegible] MSG[illegible] windows [illegible]

GetMessage [illegible] MSG [illegible] WM_QUIT [illegible]

TranslateMessage [illegible] MSG [illegible] Windows [illegible] Windows [illegible] WM_KEYDOWN [illegible] WM_KEYUP [illegible] WM_SYSKEYDOWN [illegible] WM_SYSKEYUP [illegible] WM_KEYDOWN [illegible] ASCII [illegible] TranslatMessage [illegible] ASCII [illegible] WM_CHAR [illegible] WM_SYSCHAR [illegible] ASCII [illegible] WM_CHAR [illegible] TranslateMessage [illegible]

[illegible] DispatchMessage [illegible] DispatchMessage [illegible]

[illegible]“Hello World”[illegible]“Hello world”[illegible] Windows [illegible]

[illegible]Windows[illegible]WM_PAINT

[illegible]Windows[illegible]

[illegible]

[illegible]Windows[illegible]

[illegible]

[illegible]

```
#include <windows.h>

#define CUR 0x1000    //[illegible]id

HDC hDC;//HDC[illegible]

//PAINTSTRUCT[illegible]http://
msdn.microsoft.com/en-us/library/dd162768(VS.85).aspx

//[illegible]

PAINTSTRUCT paint;

RECT rect;//RECT[illegible]



//[illegible]
```

```
LRESULT WINAPI WinProc(HWND hWnd,UINT Msg,WPARAM
wParam,LPARAM lParam)
{
    switch(Msg)//[illegible]
    {
     case WM_PAINT:
      //BginPaint [illegible] PAINSTURCT [illegible]
[illegible]
      hDC=BeginPaint(hWnd,&paint);
      //GetClientRect [illegible]
      GetClientRect(hWnd,&rect);
      //DrawText [illegible] Windows [illegible]“[illegible]”[illegible]DT_*[illegible]
[illegible]
      //[illegible]
[illegible]http://msdn.microsoft.com/en-us/library/ms901121.aspx
      //[illegible]
```

```
        DrawText(hDC,"Hello World!",-
1,&rect,DT_SINGLELINE|DT_CENTER|DT_VCENTER);
        //EndPaint □□□□□□□□□□□□
        EndPaint(hWnd,&paint);
        break;
      case WM_DESTROY://□□□□□□□□□□□□□
        PostQuitMessage(0);//□□□□□□□□□□□□□□□□□□□□□□
        return 0;//□□□□
      }
      return DefWindowProc(hWnd,Msg,wParam,lParam);
    }

    //□□□
    int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE
hPrevInstance,LPSTR lpCmdLine,int nShowCmd)
    {
      char *cName = "myWindow";
      WNDCLASSEX wc;
```

HWND hWnd;

MSG Msg;

wc.cbClsExtra = 0;

wc.cbWndExtra = 0;

wc.cbSize = sizeof(WNDCLASSEX);

wc.hbrBackground =

(HBRUSH)GetStockObject(WHITE_BRUSH);//□□□□□□□□□□□□□□□□□□

□□□□NULL□□□□□□□□

wc.hCursor =

LoadCursor(hInstance,MAKEINTRESOURCE(CUR));//□□□□□□□□□□□

□

wc.hIcon = NULL;//□□□

wc.hIconSm = NULL;//□□□

wc.hInstance = hInstance;//□□□□□□□□hInstance□□□□□□□□

wc.lpfnWndProc = WinProc;//□□□□□□□□□□□□

wc.lpszClassName =(LPSTR)cName;//□□□□□□□

wc.lpszMenuName = NULL;//□□□□□□□

```
wc.style = CS_HREDRAW | CS_VREDRAW;

RegisterClassEx(&wc);//□□□□□

hWnd = CreateWindowEx(WS_EX_CLIENTEDGE,cName,"□□□
□□□□",WS_OVERLAPPEDWINDOW,
    200,100,600,400,NULL,NULL,hInstance,NULL);//□□□□□□□
□□"□□□□□□□"
if(hWnd == NULL)
{//□□□□
    MessageBox(NULL,"There's an Error","Error
Title",MB_ICONEXCLAMATION|MB_OK);
    return 0;
}
ShowWindow(hWnd,nShowCmd);//□□□□
UpdateWindow(hWnd);

//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 while(GetMessage(&Msg,NULL,0,0))
```

```
    {
        TranslateMessage(&Msg);//□□□□
        DispatchMessage(&Msg);//□□□□
    }
    return Msg.message;
}
```

□□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□VB□JAVA□□□□□□□□□□□□□□□□□C□□□□□□□□□□□□□□□□□□□□□□□□□Windows□□Windows□□□□□□□□□□□□□□□□□□□□□Windows□□□

1.6 □□□□□□□□□

□□□ □“□□□□□”□□□□□□□□□□□□□□“□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□switch□□□□□□□□

```
case WM_LBUTTONUP://□□□□□□

    MessageBox(hWnd,"□□□□□","□□",MB_OK);

    break;

case WM_RBUTTONUP://□□□□□□

    MessageBox(hWnd,"□□□□□","□□",MB_OK);
```

break;

[illegible]

[illegible] VC 6.0 [illegible] MFC [illegible] API [illegible] API [illegible] MSDN [illegible] winuser.h [illegible]

[illegible]

[illegible] C [illegible] Windows [illegible]

1.7 [illegible]

[illegible] Windows [illegible] Windows API [illegible]

[illegible] CreateWindow [illegible] Windows [illegible] SetWindowPos [illegible] MoveWindow[illegible]

SetWindowPos

[illegible] Z [illegible] Z [illegible]

[illegible]BOOL SetWindowPos[illegible]HWN hWnd[illegible]HWND hWndInsertAfter,int X[illegible]int Y,int cx[illegible]int cy,UNIT[illegible]Flags[illegible]

[illegible]

hWnd:[illegible]

hWndInsertAfter:[illegible] z [illegible]

[illegible]

HWND_BOTTOM[illegible] Z [illegible] hWnd [illegible]

[illegible]

HWND_NOTOPMOST[illegible]

[illegible]

HWND_TOP:[illegible] Z [illegible]

HWND_TOPMOST:[illegible]

[illegible]

x[illegible]

Y[illegible]

cx:[illegible]

cy[illegible]

uFlags:[illegible]

SWP_ASNCWINDOWPOS[illegible]

[illegible]

SWP_DEFERERASE[illegible] WM_SYNCPAINT [illegible]

SWP_DRAWFRAME[illegible]

SWP_FRAMECHANGED[illegible] WM_NCCALCSIZE [illegible]

[illegible] WM_NCCALCSIZE[illegible]

SWP_HIDEWINDOW;[illegible]

SWP_NOACTIVATE[illegible]hWndlnsertAfter[illegible]

SWP_NOCOPYBITS[illegible]

SWP_NOMOVE[illegible]X[illegible]Y[illegible]

SWP_NOOWNERZORDER[illegible]z[illegible]

SWP_NOREDRAW:[illegible]

SWP_NOREPOSITION[illegible]SWP_NOOWNERZORDER[illegible]

SWP_NOSENDCHANGING[illegible]
WM_WINDOWPOSCHANGING[illegible]

SWP_NOSIZE[illegible]cx[illegible]Cy[illegible]

SWP_NOZORDER[illegible]Z[illegible]hWndlnsertAfter[illegible]

SWP_SHOWWINDOW[illegible]

[illegible]GetLastError[illegible]

[illegible]SWP_SHOWWINDOW[illegible]SWP_HIDEWINDOW[illegible]SetWindowLoog[illegible]SetWindowPos[illegible]
SWP_NOMOVEISWP_NOSIZEISWP_FRAMECHANGED[illegible]

[illegible] hWndInsertAfter [illegible]

HWND_TOPMOST [illegible] SWP_NOZORDER [illegible] Z [illegible]

[illegible]

[illegible] z [illegible]

[illegible] HWND_TOPMOST [illegible] HWND_NOTOPMOST [illegible]

[illegible] Z [illegible] hWndinsertAfter [illegible]

[illegible]

[illegible] hWndInsertAfter [illegible] HWND_NOTOPMOST [illegible]

HWND_TOPMOST [illegible]

[illegible] hWnd [illegible]

[illegible] z [illegible]

[illegible] Z [illegible]

[illegible] z [illegible]HWND_BOTTOM[illegible]

[illegible]

[illegible]

[illegible]

[illegible] SetForegroundWindow [illegible]

Windows CE[illegible] SWP_NOACTIVATE [illegible]

[illegible] SWP_NOACTIVATE [illegible]

SWP_HIDEWINDOW [illegible]

nFlags SWP_FRAMECHANGED

WindowsCE

SetwindowLong

SetWindowPos WM_WINDOWPOSCHANGED

MoveWindow

BOOL MoveWindow HWND hWnd,int x.int y,int nWidth,int nHeight,BOOL BRePaint

hWnd

x

Y

nWidth

nHaight

bRepaint: TRUE WM_PAINT

FALSE

FALSE

GetLastError

□□□□bRepaint□TRUE□□□□□□□□□□□□□□WM_PAINT□□□□□MoveWindow□□□□UPdateWindow□□□□□□bRepaint□FALSE□□□□WM_PAINT□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□WM_PAINT□□□

MoveWindow□□□□

WM_WfNOWPOSCHANGING□WM_WINDOWPOSCHANGED□WM_MOVE□WM_SIZE□WM_NCCALCSIZE□□□

□□□□□□□msdn□□□

```
#include <windows.h>


#define CUR 0x1000   //□□□□□id


HDC hDC;//HDC□□□□□□□□□□□□□□□□□□□□□□□
//PAINTSTRUCT□□□□□
//□□□□http://msdn.microsoft.com/en-us/library/dd162768(VS.85).aspx
//□□□□□□□□□□□□□
PAINTSTRUCT paint;
RECT rect;//RECT□□□□□□□□□□□□□□□□□□□□□□
```

```
//□□□□□ MoveLeft,□□□□□□□ 5 □□,□□□□□□ MoveWindow □□
int MoveLeft(HWND hWnd)
{
GetWindowRect(hWnd,&rect);//□□□□□□□
MoveWindow(hWnd,rect.left-5,rect.top,rect.right-
rect.left,rect.bottom-rect.top,TRUE);
return 1;
}

//□□□□□ MoveRight□□□□□□□□ 5 □□□□□□□□□ SetWindowPos □□
□□□□□□
int MoveRight(HWND hWnd)
{
GetWindowRect(hWnd,&rect);//□□□□□□□
SetWindowPos(hWnd,HWND_NOTOPMOST,
        rect.left+5,rect.top,
        rect.right-rect.left,
        rect.bottom-rect.top,
        SWP_NOZORDER);
return 1;
}
```

```
//[illegible]
LRESULT WINAPI WinProc(HWND hWnd,UINT Msg,WPARAM wParam,LPARAM lParam)
{
    switch(Msg)//[illegible]
    {
case WM_PAINT:
        //BginPaint [illegible] PAINSTURCT [illegible]
        hDC=BeginPaint(hWnd,&paint);
        //GetClientRect [illegible]
        GetClientRect(hWnd,&rect);
        //DrawText [illegible] Windows [illegible]“[illegible]”[illegible]DT_*[illegible]
        //[illegible]
[illegible] http://msdn.microsoft.com/en-us/library/ms901121.aspx
        //[illegible]
        DrawText(hDC,"Hello World!",-1,&rect,DT_SINGLELINE|DT_CENTER|DT_VCENTER);
        //EndPaint [illegible]
        EndPaint(hWnd,&paint);
        break;
```

```
    case WM_LBUTTONUP://□□□□□□□
        MoveLeft(hWnd);
        break;
    case WM_RBUTTONUP://□□□□□□□
        MoveRight(hWnd);
        break;
        case WM_DESTROY://□□□□□□□□□□□□
            PostQuitMessage(0);//□□□□□□□□□□□□□□□□□□□□
            return 0;//□□□□
        }
        return DefWindowProc(hWnd,Msg,wParam,lParam);
    }

    //□□□
    int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE
hPrevInstance,LPSTR lpCmdLine,int nShowCmd)
    {
        char *cName = "myWindow";
        WNDCLASSEX wc;
        HWND hWnd;
        MSG Msg;
```

```
	wc.cbClsExtra = 0;
	wc.cbWndExtra = 0;
	wc.cbSize = sizeof(WNDCLASSEX);
	wc.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);//[illegible] NULL [illegible]
	wc.hCursor = LoadCursor(hInstance,MAKEINTRESOURCE(CUR));//[illegible]
	wc.hIcon = NULL;//[illegible]
	wc.hIconSm = NULL;//[illegible]
	wc.hInstance = hInstance;//[illegible]hInstance [illegible]
	wc.lpfnWndProc = WinProc;//[illegible]
	wc.lpszClassName =(LPSTR)cName;//[illegible]
	wc.lpszMenuName = NULL;//[illegible]
	wc.style = CS_HREDRAW | CS_VREDRAW;

	RegisterClassEx(&wc);//[illegible]
```

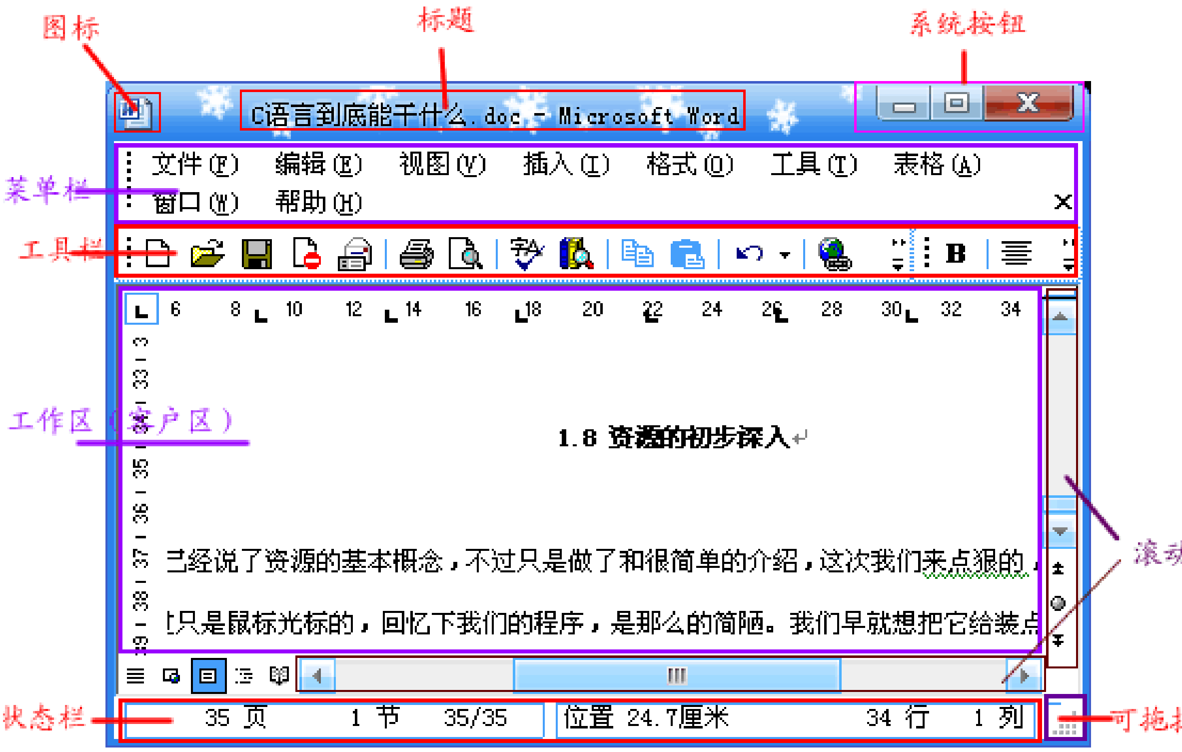
```
    hWnd =
CreateWindowEx(WS_EX_CLIENTEDGE,cName,"□□□□□
□",WS_OVERLAPPEDWINDOW,
        200,100,600,400,NULL,NULL,hInstance,NULL);//□□□□□
□□□□□"□□□□□□□"
    if(hWnd == NULL)
    {//□□□□
        MessageBox(NULL,"There's an Error","Error
Title",MB_ICONEXCLAMATION|MB_OK);
        return 0;
    }
    ShowWindow(hWnd,nShowCmd);//□□□□
    UpdateWindow(hWnd);

    //□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
  while(GetMessage(&Msg,NULL,0,0))
    {
        TranslateMessage(&Msg);//□□□□
        DispatchMessage(&Msg);//□□□□
    }
    return Msg.message;
  }
```

[illegible]

[illegible]

1.8 [illegible]

[illegible]



□□□□ Word □□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□ Vb □□□□□□□□□□□□□Java □□□□□□□□□□□□

□JFrame□□□□ IDE □□□□□□VC □□□□□□□□□□□□ VB □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□

n □□□□□□□□□□□□□□ n □□□□□□□□□□□□□□□□□□□□

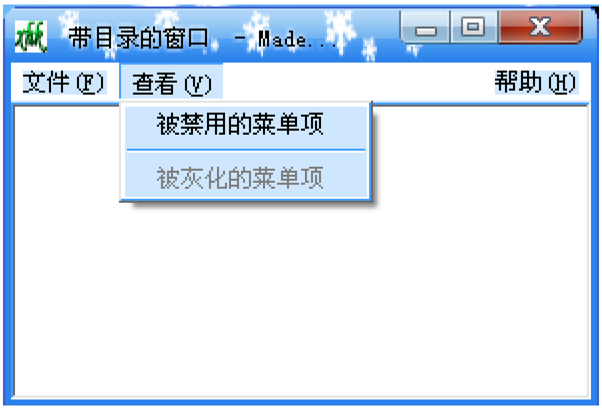
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ VC

6.0 □□□□□□□□□□□□□□□□□□ VC □ IDE □□□□□□□□□□

[illegible]VC[illegible]IDE[illegible]

[illegible]Windows[illegible]VC[illegible]

[illegible]



[illegible]“[illegible]”[illegible]

[illegible]xhk[illegible]

[illegible]“[illegible]”[illegible]

[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

[illegible]

[illegible]

1. [illegible]

□□□□□□□□□□□□□□□□□□

```
□□        ID MENU [DISCARDABLE]
BEGIN
        □□□□□□
END
```

□□□□□□□

```
□□        ID MENU [DISCARDABLE]
{
        □□□□□□
}
```

“□□ ID MENU [DISCARDABLE]”□□□□□□□□□ ID □□□□□□□□□ ID □□□ 16 □□□□□□□□□□□□□□□□□□□□□□□ ID □□□□□□□□□□□□□□□□□□ □□□□□□□□□□ ID □□□□□□□□□□□□□□□□□□□□□□□□□□ ID □□MENU □□□□□□ DISCARDABLE □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□ BEGIN □ END □□□□□□□□□□□□□□□□ {□}□□□□

□□□□□□□□□□□□□

1. □□□

MENUITEM □□□□,□□ ID [,□□□□]

2. □□□

MENUITEM SEPARATOR

3. □□□□

□□□□□□□□□□

POPUP □□□□ [,□□□□]

BEGIN

Item-definitions

END

□□□□□□□□□□

□□□□

□□□□——□□□□□□□□□□□□□□□□□□“□□□□□□□”□“□□□□□□□”□

□□ID——□□□□□□□□□□□□□□□□□□□Windows □□□□□□□□

WM_COMMAND □□□□□□□□□□□□□□ID□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□ID□

□□□□——□□□□□□□□□□□□□□□□□□□□□□

CHECHKED——□□□□□□□□□

GRAYED——□□□□□□□□□□□

INACTIVE——□□□□□□□□□□□

MENUBRREAK □ MENUBARBREAK——□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□

GREAYED——□□□

INACTIV——□□□

HELP——□□□□□□□□□□□□□□□□□□□□□□□□“□□”□□□

2.□□□□□□

□□□□□□□□□□□□□□□□□□

□□□ ID ACCELERATORS

BEGIN

□□,□□ ID[,□□][,□□]

END

BEGIN □ END □□□□□{□}□□□

□□——□□□□□□□□□□□□□□3□□□□□□

“^□□”□□□Ctrl□□□□□□□

“□□”□□□□□□□□□□□□□□VIRTKEY□

□□□□□ASCII□□□□□□□□□ASCII

□□ID——□□□□□□□Windows□□□□□□□□□ID□

□□——□□□□□□□□□□□□□□VIRTKEY□ASCII□□□□□□□“□□”□□□□□□□□□□□

ASCII□□

□□——□□□Alt,Control□Shift□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□MyMenu.rc□□□□□□□□□□□

```
/**************MyMenu.rc Written By XHK 2009.3.3*************/

#include <resource.h>

#define ICO_MAIN   0X1000        //□□

#define IDM_MAIN   0X2000        //□□

#define IDA_MAIN   0X2000        //□□□
```

```
#define IDM_OPEN 0X4101         //“□□”□□□

#define IDM_INACTIVE      0X4201         //“□□□□□□□”

#define IDM_GRAYED       0X4202         //“□□□□□□”

#define IDM_HELP  0X4301         //“□□”□□□


/********The ico file of the window***********/

ICO_MAIN   ICON     "xhk.ico"

/*******************************************/

/**Next is the definition of the Menus**********/

IDM_MAIN   menu   discardable

{

        popup  "□□(&F)"

        {

         menuitem     "□□(&O)\tCtrl+Alt+O",IDM_OPEN

        }


        popup  "□□(&V)"

        {
```

```
        menuitem    "□□□□□□□",IDM_INACTIVE,INACTIVE
        menuitem    separator
        menuitem    "□□□□□□□",IDM_GRAYED,GRAYED
       }

       popup  "□□(&H)",HELP
       {
        menuitem    "□□□□(&H)\tF1",IDM_HELP
       }
}

//□□□□□□□

IDA_MAIN accelerators
{
       VK_F1,IDM_HELP,VIRTKEY  //F1
       "O",IDM_OPEN,VIRTKEY,CONTROL,ALT   //Ctrl+Alt+O
}
```

□□□□□□□□ico□□xhk.ico□□□□□□□□□□□□□□□□□□□□□rc.exe□

MyMenu.rc□□□MyMenu.res

□□□□□□□ □□□□□□□□□□□□

```
/***********MyMenu.c      Written By XHK 2009.3.3************/

#include <windows.h>

#define ICO_MAIN  0X1000        //□□
#define IDM_MAIN  0X2000        //□□
#define IDA_MIAN  0X2000        //□□□

//□□□□
LRESULT WINAPI WinProc(HWND hWnd,UINT Msg,WPARAM
wParam,LPARAM lParam)
{
    switch(Msg)//□□□□□□□□□□□□□□□□□
    {
    case WM_DESTROY://□□□□□□□□□□□□
        PostQuitMessage(0);//□□□□□□□□□□□□□□□□□□□□
```

```
        return 0;//□□□□
    }
    return DefWindowProc(hWnd,Msg,wParam,lParam);
}

//□□□
int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE

hPrevInstance,LPSTR lpCmdLine,int


nShowCmd)
{
    char *cName = "myWindow";
         char *cCaption = "□□□□□□  - Made By XHK";
    WNDCLASSEX wc;
    HWND hWnd;
    MSG Msg;

    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;
```

```
    wc.cbSize = sizeof(WNDCLASSEX);
    wc.hbrBackground =
(HBRUSH)GetStockObject(WHITE_BRUSH);
    wc.hCursor = NULL;
    wc.hIcon =
LoadIcon(hInstance,MAKEINTRESOURCE(ICO_MAIN));//□□□□
    wc.hIconSm = NULL;
    wc.hInstance = hInstance;
    wc.lpfnWndProc = WinProc;
    wc.lpszClassName =(LPSTR)cName;
    wc.lpszMenuName = NULL;
    wc.style = CS_HREDRAW | CS_VREDRAW;

    RegisterClassEx(&wc);

    hWnd =
CreateWindowEx(WS_EX_CLIENTEDGE,cName,cCaption,WS_OVE
RLAPPEDWINDOW,
```

```
200,100,300,200,NULL,LoadMenu(hInstance,MAKEINTRESOURCE

(IDM_MAIN)),hInstance,NULL);

    if(hWnd == NULL)

    {//□□□□

        MessageBox(NULL,"There's an Error","Error

Title",MB_ICONEXCLAMATION|MB_OK);

        return 0;

    }

    ShowWindow(hWnd,nShowCmd);//□□□□

    UpdateWindow(hWnd);



     while(GetMessage(&Msg,NULL,0,0))

    {

        TranslateMessage(&Msg);//□□□□

        DispatchMessage(&Msg);//□□□□
```

```
    }
    return Msg.message;
}
```

[illegible] MyMenu.obj,z [illegible] MyMenu.res [illegible] MyMenu.exe [illegible]

[illegible]

[illegible]

1.9 [illegible]

[illegible] Windows [illegible] ID [illegible] LoadAccelerators [illegible]

LoadAccelerators

[illegible]

[illegible]HACCEL LoadAccelerators[illegible]HINSTANCE hInstance[illegible]LPCTSTR lpTableName[illegible]

[illegible]

hInstance:[illegible]

lpTableName:[illegible] MADEINTRESOURCE [illegible]

[illegible] GetLastError [illegible]

[illegible]Windows CE[illegible] RAM [illegible]

[illegible]

```
while(GetMessage(&Msg,NULL,0,0))
  {
    if(!TranslateAccelerator(hWnd,hAccel,&Msg))
    {
      TranslateMessage(&Msg);//[illegible]
      DispatchMessage(&Msg);//[illegible]
```

```
    }
  }
  return Msg.message;
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□TranslateAccelerator□□□□□□

TranslateAccelerator

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□WM_KEYDOWN□WM_SYSKEYDOWN□□□□□□□WM_COMMAND□WM_SYSCOMMAND□□□□□□□□□□□□□□□□□□□□□□□□□WM_COMMAND□WM_SYSCOMMAND□□□□□□□□□□□□□□□□

TranslateAccelerator□□□□□□□□□□□□□□□□

□□□□□int TranslateAccelerator□HWND hWnd,HACCEL hAccTable□LPMSG IpMsg□;

□□□

hWnd:□□□□□□□□□□□□□□□□

hAccTable:□□□□□□□□□□□□□□LoadAccelerators□□□□□□□□CreateAccd_eratorTable□□□□□□□

LpMsg:MSG□□□□□MSG□□□□□□□□□GetMessage□PeekMessage□□□□□□□□□□□□□□□□□□□

[illegible] GetLastError [illegible]

[illegible] WM_COMMAND [illegible] WM_SYSCOMMAND [illegible] wParam [illegible] 1[illegible] WM_SYSCOMMAND [illegible] WM_COMMAND[illegible] TransLateAccelerator [illegible] TranslateMessage [illegible] WM_INITMEMU [illegible] WM_INITMENUPOPUP [illegible]

[illegible] SetCapture [illegible] TranslatMssage [illegible] WM_SYSDEYUP [illegible] WM_SYSKEYDOWN [illegible] WM_KEYUP [illegible] WM_KEYDOWN [illegible]

[illegible] TranslateMessage [illegible] WM_COMMAND [illegible] TranslateMessage [illegible] WM_COMMAND [illegible]

[illegible]

```
/*****MyMenu.c Written by XHK 2009.3.3****/

#include <windows.h>
```

```
#define  ICO_MAIN       0X1000        //□□
#define  IDM_MAIN       0X2000        //□□
#define  IDA_MAIN       0X2000        //□□□

#define  IDM_OPEN       0X4101        //“□□”□□□
#define  IDM_INACTIVE   0X4201        //“□□□□□□□”
#define  IDM_GRAYED     0X4202        //“□□□□□□”
#define IDM_HELP        0X4301        //“□□”□□□

//□□□□
LRESULT WINAPI WinProc(HWND hWnd,UINT Msg,WPARAM
wParam,LPARAM lParam)
{
    switch(Msg)//□□□□□□□□□□□□□□□□□
    {
     case WM_COMMAND:
      switch(0x0000ffff&wParam)
      {
      case IDM_OPEN:
```

```
            MessageBox(hWnd,"□□□□\"□□\"□□□","□□",MB_OK);
            break;
          case IDM_HELP:
            MessageBox(hWnd,"□□□□\"□□□□\"□□□","□□",MB_OK);
            break;
          }
          break;
        case WM_DESTROY://□□□□□□□□□□□□
           PostQuitMessage(0);//□□□□□□□□□□□□□□□□□□□□
           return 0;//□□□□
        }
        return DefWindowProc(hWnd,Msg,wParam,lParam);
    }

    //□□□
    int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE
hPrevInstance,LPSTR lpCmdLine,int nShowCmd)
    {
```

```
char *cName = "myWindow";
 char *cCaption = "□□□□□□  - Made By XHK";
WNDCLASSEX wc;
HWND hWnd;
 HACCEL hAccel;//□□□□□□
MSG Msg;

wc.cbClsExtra = 0;
wc.cbWndExtra = 0;
wc.cbSize = sizeof(WNDCLASSEX);
wc.hbrBackground =
(HBRUSH)GetStockObject(WHITE_BRUSH);
wc.hCursor = NULL;
wc.hIcon =
LoadIcon(hInstance,MAKEINTRESOURCE(ICO_MAIN));//□□□□
wc.hIconSm = NULL;
wc.hInstance = hInstance;
wc.lpfnWndProc = WinProc;
```

```
        wc.lpszClassName =(LPSTR)cName;

        wc.lpszMenuName = NULL;

        wc.style = CS_HREDRAW | CS_VREDRAW;


        RegisterClassEx(&wc);


        hWnd =

    CreateWindowEx(WS_EX_CLIENTEDGE,cName,cCaption,WS_OVE

    RLAPPEDWINDOW,



    400,300,300,200,NULL,LoadMenu(hInstance,MAKEINTRESOURCE

    (IDM_MAIN)),hInstance,NULL);

        if(hWnd == NULL)

        {//□□□□

            MessageBox(NULL,"There's an Error","Error

    Title",MB_ICONEXCLAMATION|MB_OK);

            return 0;

        }
```

```
ShowWindow(hWnd,nShowCmd);//□□□□
UpdateWindow(hWnd);

//□□□□□□□

hAccel=LoadAccelerators(hInstance,MAKEINTRESOURCE(IDA_MA
IN));

while(GetMessage(&Msg,NULL,0,0))
{
   //□□□□□□□□□□□
  if(!TranslateAccelerator(hWnd,hAccel,&Msg))
  {
     TranslateMessage(&Msg);//□□□□
     DispatchMessage(&Msg);//□□□□
  }
}
return Msg.message;
```

}

[illegible]

1.10 [illegible]

[illegible]

[illegible] API [illegible]

1.[illegible]

BOOL AppendMenu□hMenu hMenu□UINT uFlags□UINT uIDNewltem□LPCTSTR lpNewltem□;

2.[illegible]

BOOL InsertMenu(HMENU hMenu,UINt uPosition,UINT uFlags,UINT uIDNewltem,LPCTSTR lpNewltem);

3.[illegible]

BOOL ModifyMenu(HMENU hMnu,UINT uPosition,UINT uFlags,UINT uIDNewItem,LPCTSTR lpNewItem);

4.□□□□

BOOL DeleteMenu(HMENU *hMenu*,UINT *uPosition*,UINT *uFlags*);

BOOL RemoveMenu(HMENU *hMenu*, UINT *uPosition*, UINT *uFlags*);

□□ AppendMenu □ InsertMenu □□□□□□□□□□□□ ApendMenu □□□□□□□□□□ InsertMenu □□□□□□□□□□□ DeleteMenu □ RemoveMenu □□□□□□□□□□□□□□□□□□ popup □□□□□□ DeleteMenu □□□□□□□□□□ popup □□□□□□□□□□□□ RemoveMenu □□□□□□□□□ popup □□□□□ popup □□□□□□□□□□

□□□□□□□□□□□□ hMenu □□□□□□□□□ uPositon □□□□□□□□□ □□□□□□□□□ ID □□□□□□□□□□□□□□□□□□□ uFlags □□□□ uFlags □ MF_BYCOMMAND □□ uPosition □□□□□□ ID□□ uFlags □ MF_BYPOSITON □□ uPosition □□□□□□□□□□□□ 0 □□□□□□□□□□□ 0.

AppendMenu □ InsertMenu □□□ uIDNewItem □□□□□□□□□

ID□lpNewmItem □□□□□□□□□□□ModifyMenu □□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□ TrackPopupMenu □□□

TrackPopupMenu

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□BOOL TrackPopupMenu□HMENU hMenu□UINT uFlags□int x□int y□int nReserved□HWND hWnd□CONST RECT”prcRect□□

□□

hMenu□□□□□□□□□□□□□□□□ CreatePopupMenu □□□□□□□□□□□□□□□ GetSubMenu □□□□□□□□□□□□□□□□□□□

uFlags□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

TPM_CENTERALIGN□□□□□□□□□□□ x □□□□□□□□□□□□□

TPM_LEFTALIGN□□□□□□□□□□□□□□□□□□□ X □□□□□□

TPM_RIGHTALIGN□□□□□□□□□□□□□□□□□□□ X □□□□□□□

□□□□□□□□□□□□□□□□□□□□

TPM_BOTTOMALIGN□□□□□□□□□□□□□□□□□□□ y □□□□□□□

TPM_TOPALIGN□□□□□□□□□□□□□□□□□□□ y □□□□□□□

TPM_VCENTERALIGN□□□□□□□□□□y□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□

TPM_NONOTIFY□□□□□□□□□□□□□□□□□□□

TPM_RETURNCMD□□□□□□□□□□□□□□□□□□□□□

(□□□TrackPopupMenu□□□□0□□□□□□□□□□□□□□□□□□ID□□□wParam□□□□□□□□□WM_SYSCOMMAND□□)

□□□□□□□□□□□□□□□□□□

TPM_LEFTBUTTON:□□□□□□□□□□□□□□□□

TPM_RIGHTBUTTON□□□□□□□□□□□□□□□□□

X□□□□□□□□□□□□□

Y:□□□□□□□□□□□□

NReserved□□□□□□□

HWnd□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□WM_COMMAND□□

□□□□uFlags□□□TPM_NONOTIFY□□□□□hWnd□□□□□□□ □□□hWnd□□□□□□□□□□□□□□□□□□□

PrcRect□□□

□□□□□□uFlags□□□TPM_RETURNCMD□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□uFlags□□TPM_RETURNCMD□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□GetLastError

□□:

□□□Windows CE □□□□□ uFlags □□□□□TPM_NONOTIFY□TPM_LEFTBUTTON□TPM_RIGHTBUTTON□

□□□□□□□□□□□□□□□□□□

GetCursorPos

□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□BOOL GetCursorPos□LPPOlNT IpPoint□□

□□□

IpPint□POINT □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□ Public Type POINTAPI

x As Long

y As Long

End Type

GetCursorPos biao

□□ biao.x □□□□□□□□□ x □□□,biao.y □□□□□□ y □□□□□

□□GetLastError □□□

□□ TrackPopupMenu □□□□□□□□□□□□□□□ popup □□□□□□LoadMenu □□□□□□□□ popup □□□□popup □□□□□□□□□□□□□□□□□□□

GetSubMenu □□□□□□□□□□□□□□□□ popup □□□□□□□GetSubMenu □□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□□MyMenu.rc □□□□□□□□

```
/**************MyMenu.rc Written By XHK
2009.3.3*************/

/*****MyMenu.c Modified by XHK 2009.3.4****/

#include <resource.h>

#define  ICO_MAIN      0X1000     //□□
#define  IDM_MAIN      0X2000     //□□
#define  IDA_MAIN      0X2000     //□□□

#define  IDM_OPEN      0X4101     //“□□”□□□
#define  IDM_INACTIVE  0X4201     //“□□□□□□□”
#define  IDM_GRAYED    0X4202     //“□□□□□□”
#define IDM_HELP       0X4301     //“□□”□□□

#define IDM_SHORTCUT   0X4400     //□□□□
```

```
#define IDM_APPEND          0X4401       //“□□□□□□”

#define IDM_DELETE 0X4402       //“□□□□□”


/********The ico file of the window***********/

ICO_MAIN ICON  "xhk.ico"

/*******************************************/

/**Next is the definition of the Menus**********/

IDM_MAIN menu discardable

{

    popup  "□□(&F)"

    {

     menuitem    "□□(&O)\tCtrl+Alt+O",IDM_OPEN

    }



    popup  "□□(&V)"

    {

     menuitem    "□□□□□□□",IDM_INACTIVE,INACTIVE

     menuitem    separator
```

```
     menuitem    "□□□□□□□",IDM_GRAYED,GRAYED
    }

    popup  "□□(&H)",HELP
    {
     menuitem    "□□□□(&H)\tF1",IDM_HELP
    }
}

IDM_SHORTCUT        menu          discardable
{
    popup"    "//□□□□□□□□□□□□
    {
     menuitem   "□□□□□□",IDM_APPEND
     menuitem   "□□□□□",IDM_DELETE
    }
}

//□□□□□□□
```

```
IDA_MAIN accelerators
{
    VK_F1,IDM_HELP,VIRTKEY  //F1
    "O",IDM_OPEN,VIRTKEY,CONTROL,ALT   //Ctrl+Alt+O
}
```

□□□□□□□□□□□□

```
/*****MyMenu.c Written by XHK 2009.3.3****/
/*****MyMenu.c Modified by XHK 2009.3.4****/
#include <windows.h>

#define  ICO_MAIN       0X1000      //□□
#define  IDM_MAIN       0X2000      //□□
#define  IDA_MAIN       0X2000      //□□□

#define  IDM_OPEN       0X4101      //“□□”□□□
#define  IDM_INACTIVE   0X4201      //“□□□□□□□”
#define  IDM_GRAYED     0X4202      //“□□□□□□”
#define IDM_HELP        0X4301      //“□□”□□□
```

```
#define IDM_SHORTCUT        0X4400
#define IDM_APPEND   0X4401
#define IDM_DELETE   0X4402

HMENU hMenu;//[illegible]
HMENU  hMenuShort;//[illegible]

//[illegible]
int PopupShortcutMenu(hWnd)
{
    POINT point;
    GetCursorPos(&point);//[illegible]
    //[illegible]

TrackPopupMenu(GetSubMenu(hMenuShort,0),TPM_CENTERALIG
N,point.x,point.y,0,(HWND)hWnd,NULL);
    return 1;
}
```

```
//□□□□
LRESULT WINAPI WinProc(HWND hWnd,UINT Msg,WPARAM
wParam,LPARAM lParam)
{
    //HMENU hMenu;
     //HINSTANCE hInstance;
     switch(Msg)
    {
     case WM_COMMAND:
      switch(0x0000ffff&wParam)
      {
      case IDM_OPEN:
         MessageBox(hWnd,"□□□□\"□□\"□□□","□□",MB_OK);
         break;
      case IDM_HELP:
         MessageBox(hWnd,"□□□□\"□□□□\"□□□","□□",MB_OK);
         break;
      case IDM_APPEND:
```

```
        //□□□□□□

AppendMenu(GetSubMenu(hMenuShort,0),MF_CHECKED,
0X4403,"□□□□□□□");
        break;
      case IDM_DELETE:
        //□□□□□

DeleteMenu(GetSubMenu(hMenuShort,0),0x4403,MF_BYC
OMMAND);
        break;
      }
      break;
     case WM_RBUTTONUP:
      PopupShortcutMenu(hWnd);
```

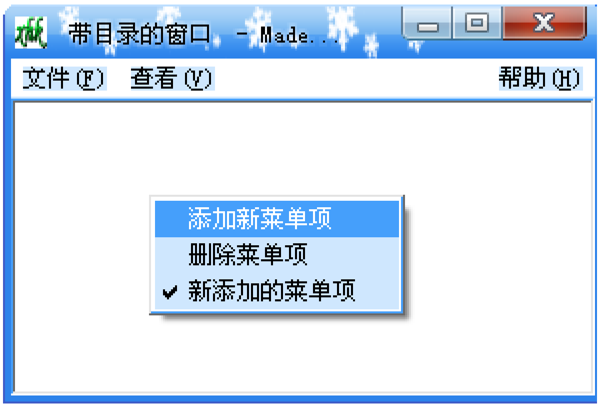
```
      break;
    case WM_DESTROY://□□□□□□□□□□
      PostQuitMessage(0);
      return 0;//□□□□
    }
    return DefWindowProc(hWnd,Msg,wParam,lParam);
  }

  //□□□
  int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE
hPrevInstance,LPSTR lpCmdLine,int nShowCmd)
  {
    char *cName = "myWindow";
     char *cCaption = "□□□□□□  - Made By XHK";
    WNDCLASSEX wc;
    HWND hWnd;

     HACCEL hAccel;//□□□□□□
```

```
MSG Msg;

wc.cbClsExtra = 0;
wc.cbWndExtra = 0;
wc.cbSize = sizeof(WNDCLASSEX);
wc.hbrBackground =
(HBRUSH)GetStockObject(WHITE_BRUSH);
wc.hCursor = NULL;
wc.hIcon =
LoadIcon(hInstance,MAKEINTRESOURCE(ICO_MAIN));//□□□□
wc.hIconSm = NULL;
wc.hInstance = hInstance;
wc.lpfnWndProc = WinProc;
wc.lpszClassName =(LPSTR)cName;
wc.lpszMenuName = NULL;
wc.style = CS_HREDRAW | CS_VREDRAW;

RegisterClassEx(&wc);
```

```
        hMenuShort =
LoadMenu(hInstance,MAKEINTRESOURCE(IDM_SHORTCUT));
        hMenu =
LoadMenu(hInstance,MAKEINTRESOURCE(IDM_MAIN));
       hWnd =
CreateWindowEx(WS_EX_CLIENTEDGE,cName,cCaption,WS_OVE
RLAPPEDWINDOW,
           400,300,300,200,NULL,hMenu,hInstance,NULL);
       if(hWnd == NULL)
       {//□□□□
           MessageBox(NULL,"There's an Error","Error
Title",MB_ICONEXCLAMATION|MB_OK);
           return 0;
       }
       ShowWindow(hWnd,nShowCmd);//□□□□
       UpdateWindow(hWnd);
```

```
//□□□□□□□

hAccel=LoadAccelerators(hInstance,MAKEINTRESOURCE(IDA_MA
IN));

while(GetMessage(&Msg,NULL,0,0))
{
    //□□□□□□□□□□□
    if(!TranslateAccelerator(hWnd,hAccel,&Msg))
    {
        TranslateMessage(&Msg);//□□□□
        DispatchMessage(&Msg);//□□□□
    }
}
return Msg.message;
}
```

□□□□□□□□□□□□□□

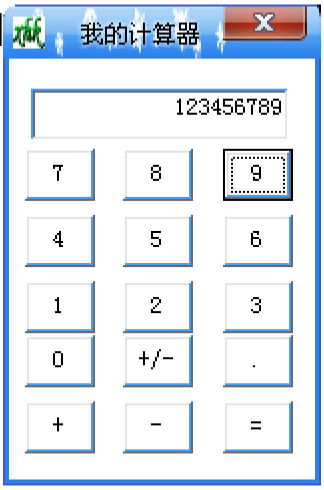


[illegible] Ctrl+C[illegible]Ctrl+V [illegible]

1.11 [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] MessageBox [illegible]

[illegible]“[illegible]”[illegible]

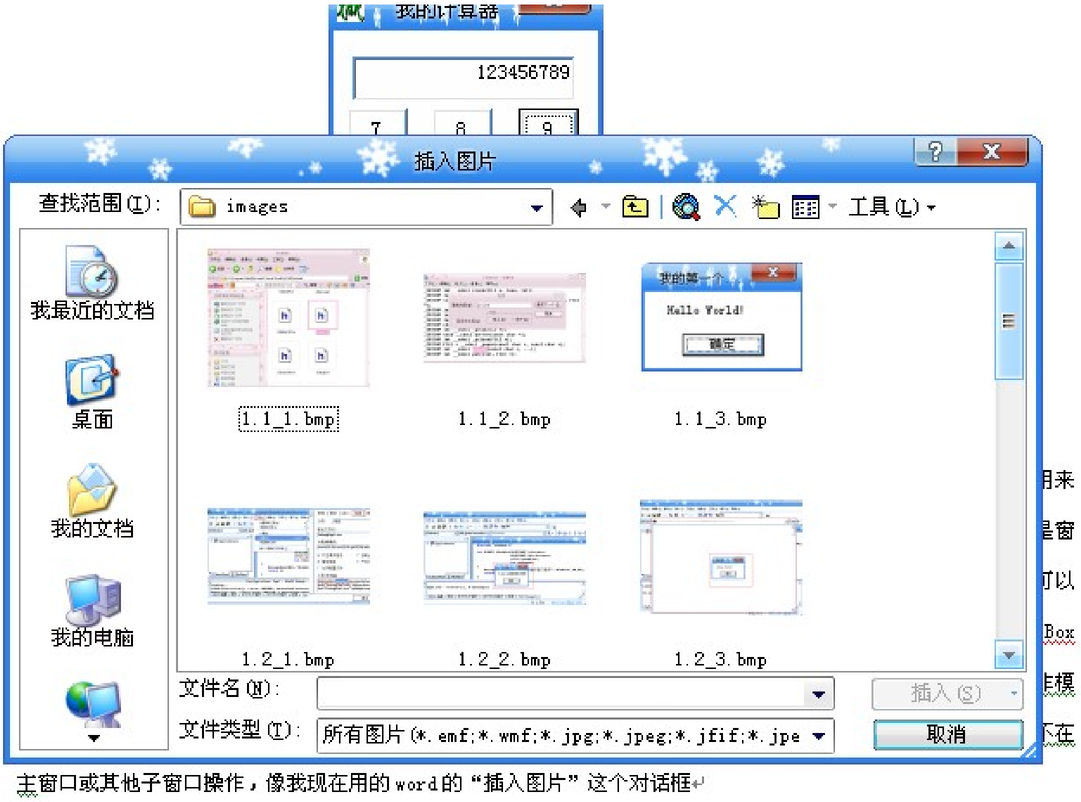
[illegible]

[illegible] Windows [illegible]

MessageBox [illegible]

[illegible]

[illegible] word [illegible]“[illegible]”[illegible]



[illegible] word [illegible]

[illegible]“[illegible]”[illegible]

[illegible] DialogBoxParam [illegible] CreateDialogParam [illegible]

Windows [illegible] CreateWindowEx [illegible]

[illegible] Windows [illegible]

“[illegible]”[illegible]Windows [illegible]

[illegible]

[illegible]

[illegible]

Windows [illegible]Windows [illegible]

[illegible]

[illegible]Windows [illegible]

DialogBoxParam [illegible] CreatDialogParam [illegible]

[illegible]

[illegible] DialogBoxParam [illegible]

[illegible]

[illegible]

[illegible]

[illegible] ID DIALOG [DISCARDABLE] x [illegible],y [illegible],[illegible],[illegible] [[illegible]]

{

　　□□□□□

}

□□□□□□□

□□□□ CAPTION “□□” □□□□□□□□□□□□□□

□□□ CLASS“□□” □□□□□□□□□□□□□□□□□□□□□Windows□□□□

□□□□ STYLE □□□□ □□□□□□□□

□□□□ EXSTYLE □□□□ □□□□□□□□□□

□□ FONT □□ “□□□” □□□□□□□□□□□□□

□□ MENU □□ID □□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Windows□□□

□□□□□□□□□□

CONTROL □□,ID,□□,x,y,□□,□□,[,□□□□]

□

CONTROL [□□,] ID□x,y,□□,□□,[,□□][,□□□□]

□□□□□□□□□PUSHBUTTON□□□□□□□DEEPBUTTON□□□□□

□CHECKBOX□□□□□□□RADIOBUTTON□□□□□EDITTEXT□□□□□

□COMBOBOX□□□□□□LISTBOX□□□□□□GROUPBOX□□□□□

□SCROLLBAR□□□□□□ICON□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ msdn □□□

□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□

```
/***********MyCalculator.rc**************/

#include <resource.h>

#define ICO_MAIN  0X1000
#define DLG_MAIN 1

#define IDB_0            0X4400    //0
#define IDB_1            0X4401    //1
```

```
#define IDB_2          0X4402     //2

#define IDB_3          0X4403     //3

#define IDB_4          0X4404     //4

#define IDB_5          0X4405     //5

#define IDB_6          0X4406     //6

#define IDB_7          0X4407     //7

#define IDB_8          0X4408     //8

#define IDB_9          0X4409     //9

#define IDB_PLUS   0X4410     //+

#define IDB_SUB        0X4411     //-

#define IDB_EQU        0X4412     //=

#define IDB_DOT        0X4413     //.

#define IDB_PAS        0X4414     //□□□

#define IDB_EDIT   0x4415     //□□□


ICO_MAIN  ICON     "xhk.ico"


DLG_MAIN DIALOG 300,150,102,140   STYLE  DS_SETFONT |

DS_MODALFRAME | WS_CAPTION | WS_SYSMENU
```

```
CAPTION    "□□□□□"
FONT 9,"□□"
{
        //□□□□□□□
        EDITTEXT IDB_EDIT,7,10,86,17,ES_RIGHT

        PUSHBUTTON "7",IDB_7,5,30,23,17
        PUSHBUTTON "8", IDB_8, 38,30,23,17
        PUSHBUTTON "9",IDB_9,71,30,23,17

        PUSHBUTTON "4",IDB_4,5,52,23,17
        PUSHBUTTON "5",IDB_5,38,52,23,17
        PUSHBUTTON "6",IDB_6,71,52,23,17

        PUSHBUTTON "1",IDB_1,5,74,23,17
        PUSHBUTTON "2",IDB_2,38,74,23,17
        PUSHBUTTON "3",IDB_3,71,74,23,17

        PUSHBUTTON "0",IDB_0,5,92,23,17
        PUSHBUTTON "+/-",IDB_PAS,38,92,23,17
```

```
    PUSHBUTTON ".",IDB_DOT,71,92,23,17

    PUSHBUTTON "+",IDB_PLUS,5,114,23,17
    PUSHBUTTON "-",IDB_SUB,38,114,23,17
    PUSHBUTTON "=",IDB_EQU,71,114,23,17
}
```

□□□□□□□□□□

```
/***********MyCalculator.c**************/

#include <windows.h>
#include <stdio.h>

#define ICO_MAIN  0X1000
#define DLG_MAIN 1

#define IDB_0          0X4400     //0
#define IDB_1          0X4401     //1
#define IDB_2          0X4402     //2
#define IDB_3          0X4403     //3
#define IDB_4          0X4404     //4
```

```
#define IDB_5			0X4405		//5
#define IDB_6			0X4406		//6
#define IDB_7			0X4407		//7
#define IDB_8			0X4408		//8
#define IDB_9			0X4409		//9
#define IDB_PLUS	0X4410		//+
#define IDB_SUB			0X4411		//-
#define IDB_EQU			0X4412		//=
#define IDB_DOT			0X4413		//.
#define IDB_PAS			0X4414		//□□□
#define IDB_EDIT	0x4415		//□□□


int  num1=0;//□□□□□□□□
int  num2=0;//□□□□□□□□
char s[10];//□□□□□□□□□□□□□□□□□□□□□□□□□□□□
UINT uFlags=1;//□□□□□□□□□□□□□□□□□num1□num2□□
char oPration='+';//□□□□□□□□□□□□□□□□□□□□□□
```

```
//□□□□□□□□□

int StrToNum(char * str)

{

        return  atoi(str);

}


//□□□□□□□□□

char * NumToStr(int nNum)

{



        itoa(nNum,s,10);

        return s;



}


//□□□□□□□□□□

int SetEditValue(int nNum,HWND hEdit)

{

        if(uFlags==1)
```

```
        {
         num1 = num1*10+nNum;//□□□□□□□□□□□□□
         SetWindowText(hEdit,NumToStr(num1));
        }
        else
        {
         num2 = num2*10+nNum;//□□□□□□□□□□□□□
         SetWindowText(hEdit,NumToStr(num2));
        }
        return 0;
}

LRESULT WINAPI DialogProc(HWND hWnd,UINT Msg,WPARAM
wParam,LPARAM lParam)
{
        HICON hIcon;
        HWND hEdit = GetDlgItem(hWnd,IDB_EDIT);
        switch(Msg)
```

```
{
case WM_INITDIALOG:
 //□□□□
 hIcon =
LoadIcon(GetModuleHandle("MyCalculator.exe"),MAKEINTRESOU
RCE(ICO_MAIN));
 SendMessage(hWnd,WM_SETICON,ICON_BIG,(long)hIcon);
 SetWindowText(hEdit,"0.");//□□□□□□□□□□"0."
 return TRUE;
case WM_COMMAND:
 switch(LOWORD(wParam))//LOWORD(wParam)□□□□□□ID
 {
 //□□□□□□□□□□□□□□□□□□□□□□□□□□
 //□□□□□□□□□□□□□□□□□□□□
 case IDB_0:
    SetEditValue(0,hEdit);
    break;
 case IDB_1:
```

```
    SetEditValue(1,hEdit);
    break;
case IDB_2:
    SetEditValue(2,hEdit);
    break;
case IDB_3:
    SetEditValue(3,hEdit);
    break;
case IDB_4:
    SetEditValue(4,hEdit);
    break;
case IDB_5:
    SetEditValue(5,hEdit);
    break;
case IDB_6:
    SetEditValue(6,hEdit);
    break;
case IDB_7:
```

```
    SetEditValue(7,hEdit);
    break;
case IDB_8:
    SetEditValue(8,hEdit);
    break;
case IDB_9:
    SetEditValue(9,hEdit);
    break;
case IDB_PLUS:
    oPration = '+';
    uFlags=0;
    break;
case IDB_SUB:
    oPration = '-';
    uFlags=0;
    break;
case IDB_DOT:
    MessageBox(hWnd,"□□□□□□□□□□□□□□","□□",MB_OK);
```

```
    break;
  case IDB_PAS:
    MessageBox(hWnd,"□□□□□□□□□□□□□□","□□",MB_OK);
    break;
  case IDB_EQU:
    //□□□□□□□□□□□□num1□num2□□□□□□□□□□□
    if(oPration=='+')
        SetWindowText(hEdit,NumToStr(num1+num2));
    else
        SetWindowText(hEdit,NumToStr(num1-num2));
    uFlags=1;
    num1=0;
    num2=0;
    break;
  }


  break;
 case WM_CLOSE://□□□□
```

```
        EndDialog(hWnd,0);//□□□□□□□

        return TRUE;

       }

       return FALSE;

}

int WINAPI WinMain(HINSTANCE hInstance,

                   HINSTANCE hPrevInstance,

                   LPSTR lpCmdLine,

                   int nShowCmd)

{

       //□□□□□□□□□


DialogBoxParam(hInstance,MAKEINTRESOURCE(DLG_MAIN),NULL

,DialogProc,0);

       return 1;

}
```

□□□QQ□□□□□□□□

1.12 □□QQ□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□QQ□□□□□□□□□□□□□□□□□□□□□□QQ□□□□□□□□□□□□QQ 2009□□□□□□□□□□□

[illegible]

[illegible]

[illegible]

[illegible] QQ [illegible]

[illegible]

```
/*************MyqqWnd.rc Written By XHK****************/

#include <resource.h>

#define DLG_MAIN 1
#define ICO_MAIN   0X1000
#define IDB_QQ              2
```

```
#define IDE_USER   0X4101        //□□□□□

#define IDE_PASS   0X4102        //□□□□□


#define IDC_RECORD       0X4201       //□□□□□□□

#define IDC_AUTO   0X4202        //□□□□□□□


#define IDB_CHECK        0X4301       //□□□□□□

#define IDB_SET          0X4302       //□□□□

#define IDB_SUBMIT       0X4303       //□□□□


#define IDC_QQ     0X4304


ICO_MAIN ICON      "qq.ico"

IDB_QQ BITMAP      "qq.bmp"


DLG_MAIN DIALOG 255,205,222,145

STYLE WS_SYSMENU | WS_MINIMIZEBOX

CAPTION "QQ 2009 Made By XHK"

FONT 9,"□□"

{
```

```
CONTROL "AAAAA",IDC_QQ,"Static",SS_BITMAP | WS_CHILD

|

WS_VISIBLE,0,0,20,40

GROUPBOX "",-1,-1,48,224,75

RTEXT "□□:",IDC_STATIC,14,55,20,15,SS_CENTERIMAGE

EDITTEXT IDE_USER,37,55,120,15,ES_NUMBER

LTEXT "□□□□□",-1,160,55,50,15,SS_CENTERIMAGE


RTEXT "□□:",-1,14,75,20,15,SS_CENTERIMAGE

EDITTEXT IDE_PASS,37,75,120,15,ES_PASSWORD

LTEXT "□□□□",-1,160,75,45,15,SS_CENTERIMAGE

CHECKBOX "□□□□",IDC_RECORD,20,105,50,15

CHECKBOX "□□□□",IDC_AUTO,100,105,50,15

PUSHBUTTON "□□□□",IDB_CHECK,5,125,50,15

PUSHBUTTON "□□",IDB_SET,60,125,50,15
```

```
        DEFPUSHBUTTON "□□",IDB_SUBMIT,150,125,50,15
}
```

□□□□□□□□□□□□□□□

```
#include <windows.h>



#define DLG_MAIN 1
#define ICO_MAIN    0X1000
#define IDB_QQ          2

#define IDE_USER    0X4101
#define IDE_PASS    0X4102

#define IDC_RECORD      0X4201
#define IDC_AUTO    0X4202

#define IDB_CHECK       0X4301
#define IDB_SET         0X4302
#define IDB_SUBMIT      0X4303
```

```
#define IDC_QQ      0X4304



LRESULT WINAPI DialogProc(HWND hWnd,UINT Msg,WPARAM
wParam,LPARAM lParam)
{
        HICON hIcon;
        HWND  hImage;
        HBITMAP hBitmap;
        HINSTANCE hInstance;
        PAINTSTRUCT   ps;
        HDC   hDC;
        RECT   rc;
        switch(Msg)
        {
        case WM_INITDIALOG:
         //□□□□□□□
```

```
        hInstance = GetModuleHandle(NULL);

        hIcon =

LoadIcon(hInstance,MAKEINTRESOURCE(ICO_MAIN));

        SendMessage(hWnd,WM_SETICON,ICON_BIG,(long)hIcon);

        //□□□□□□ qq.bmp

        hBitmap =

LoadBitmap(hInstance,MAKEINTRESOURCE(IDB_QQ));

        hImage = GetDlgItem(hWnd,IDC_QQ);

        SendMessage(hImage,STM_SETIMAGE,IMAGE_BITMAP,

(long)hBitmap);


        break;

       case WM_PAINT:

        //□□□□□□□□□□□□□□

        GetClientRect(hWnd,&rc);

        hDC   =   BeginPaint(hWnd,&ps);
```

```
        FillRect(hDC,&rc,(HBRUSH)CreateSolidBrush(RGB(200,
227, 255)));

//□□□□□
        EndPaint(hWnd,&ps);
        return   0;
       case WM_CLOSE:
        EndDialog(hWnd,0);
        return TRUE;
       }
       return FALSE;
}


int WINAPI WinMain(HINSTANCE hInstance,
                HINSTANCE hPreInstance,
                LPSTR lpCmdLine,
                int                 nShowCmd)
{
```

```
DialogBoxParam(hInstance,MAKEINTRESOURCE(DLG_MAIN),NULL
,DialogProc,0);

	return 1;
}
```

□□qq.bmp□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Windows□□□□□□□□□□□□□CreatWindows□□

1.3 □□□

□□QQ□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

QQ [illegible] [illegible]

[illegible]

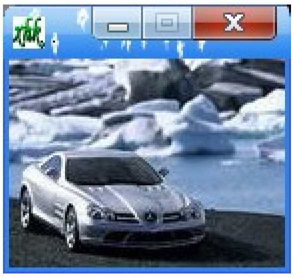
Microsoft Windows [illegible] [illegible]Windows [illegible]

[illegible] Windows [illegible] SetTimer [illegible] Windows [illegible] [illegible]“[illegible]”[illegible] SetTimer [illegible] [illegible] WM_TIMER [illegible] 1000ms[illegible]1s[illegible] Windows [illegible] WM_TIMER [illegible] 1~4,294,967,295ms[illegible] 50 [illegible] KillTimer [illegible]

[illegible] WM_TIMER [illegible]Windows [illegible] WM_TIMER[illegible] WM_TIMER[illegible]

WM_TIMER [illegible] WM_TIMER [illegible] WM_TIMER [illegible] WM_TIMER [illegible]

[illegible]



[illegible]

```
/**************Timer.rc**************/

#include <resource.h>

#define DLG_MAIN 1

#define ICO_MAIN 0X1000

#define IDB_BMP1 0X4101
```

```
#define IDB_BMP2 0x4102

#define IDC_BMP  0x4103


ICO_MAIN ICON   "XHK.ICO"

IDB_BMP1 BITMAP "1.BMP"

IDB_BMP2 BITMAP "2.BMP"


DLG_MAIN DIALOG 255,205,92,69

STYLE WS_SYSMENU | WS_MINIMIZEBOX

CAPTION "Timer"

FONT 9,"□□"

{

        CONTROL "AAAAA",IDC_BMP,"Static",SS_BITMAP |

WS_CHILD | WS_VISIBLE,0,0,20,40

}
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□msdn□□□□□□□□□□□□□□□□□□□□□

```
/****************Timer.c*************/
```

```
#include <windows.h>

#define DLG_MAIN 1
#define IDB_BMP1 0X4101
#define IDB_BMP2 0x4102
#define IDC_BMP  0x4103
#define ICO_MAIN 0x1000

UINT uFlags=1;

LRESULT CALLBACK DialogProc(HWND hWnd,UINT
uMsg,WPARAM wParam,LPARAM lParam)
{
	HINSTANCE hInstance = GetModuleHandle(NULL);
	HBITMAP   hBitmap;
	HWND      hIdc = GetDlgItem(hWnd,IDC_BMP);
	HICON    hIcon;

	switch(uMsg)
```

```
    {
    case WM_INITDIALOG:
      hBitmap =
LoadBitmap(hInstance,MAKEINTRESOURCE(IDB_BMP1));

      hIcon =
LoadIcon(hInstance,MAKEINTRESOURCE(ICO_MAIN));
      SendMessage(hWnd,WM_SETICON,ICON_BIG,(long)hIcon);
      SendMessage(hIdc,STM_SETIMAGE,IMAGE_BITMAP,
(long)hBitmap);
      SetTimer(hWnd,0,500,NULL);//□□□□□
      return TRUE;
    case WM_TIMER:
      //□□□□□□□□□□□□□
      if(uFlags)
         hBitmap =
LoadBitmap(hInstance,MAKEINTRESOURCE(IDB_BMP1));
```

```
            else

               hBitmap =

LoadBitmap(hInstance,MAKEINTRESOURCE(IDB_BMP2));

            SendMessage(hIdc,STM_SETIMAGE,IMAGE_BITMAP,

(long)hBitmap);

            uFlags = uFlags ^ 1;//□□□□□□□1□□□0□□□0□□□1

            return TRUE;

        case WM_CLOSE:

            KillTimer(hWnd,0);//□□□□□

            EndDialog(hWnd,0);

            return TRUE;

        }

        return FALSE;

}


int WINAPI WinMain(HINSTANCE hInstance,

                   HINSTANCE hPreInstance,

                   LPSTR  lpCmdline,
```

```
                    int                nShowCmd)
{


DialogBoxParam(hInstance,MAKEINTRESOURCE(DLG_MAIN),0,DialogProc,0);
        return 1;
}
```

[illegible] bmp [illegible] OK [illegible]

1.14 [illegible]——[illegible]

[illegible] MoveWindow [illegible]

□□

□□□□□□□□□□□

```
/*******************WobbleWnd.rc***************/

#include <resource.h>

#define DLG_MAIN 1
#define ICO_MAIN 0X1000

//#define IDB_BMP   0X4000

ICO_MAIN ICON "XHK.ICO"
//IDB_BMP  BITMAP "XHK.BMP"

DLG_MAIN DIALOG 255,205,100,100
STYLE WS_SYSMENU | WS_MINIMIZEBOX
CAPTION "Wobble Window"
FONT 9,"□□"
{
```

}

[illegible]

```c
/***********WobbleWnd.c*****************/

#include <windows.h>

#define DLG_MAIN 1
#define ICO_MAIN 0X1000


LRESULT CALLBACK DialogProc(HWND hWnd,UINT
uMsg,WPARAM wParam,LPARAM lParam)
{
        HINSTANCE hInstance = GetModuleHandle(NULL);
        HICON     hIcon;
        HWND      hDestWnd;

        RECT    rc;
```

```
int       i=0;


switch(uMsg)

{

case WM_INITDIALOG:

  hIcon =

LoadIcon(hInstance,MAKEINTRESOURCE(ICO_MAIN));

  SendMessage(hWnd,WM_SETICON,ICON_BIG,(long)hIcon);

  SetTimer(hWnd,0,10,NULL);

  return TRUE;

case WM_TIMER:


  i=(rand()-rand())/1000;


  if((hDestWnd=GetForegroundWindow())!=NULL){

  GetWindowRect(hDestWnd,&rc);

  if(rc.top<0||rc.bottom>800)

     rc.top=100;
```

```
            if(rc.left<0||rc.right>1000)
                rc.left=100;

            MoveWindow(hDestWnd,rc.left+i,rc.top+i,rc.right-
rc.left,rc.bottom-rc.top,TRUE);
            }
            //SendWindowMessage(
            return TRUE;
         case WM_CLOSE:
            KillTimer(hWnd,0);//□□□□□
            EndDialog(hWnd,0);
            return TRUE;
         }
         return FALSE;
}

int WINAPI WinMain(HINSTANCE hInstance,
                   HINSTANCE hPreInstance,
```

```
                    LPSTR  lpCmdline,
                    int                 nShowCmd)
{

DialogBoxParam(hInstance,MAKEINTRESOURCE(DLG_MAIN),0,Dia
logProc,0);
        srand((unsigned)time(NULL));
        return 1;
}
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Come on